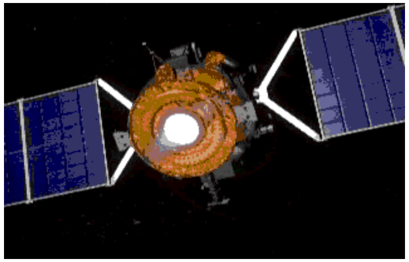
Explanation of Accomplishment

- **POC:** Klaus Havelund and Willem Visser
  (ASE group, Code IC, {havelund,wvisser}@email.arc.nasa.gov)
- **Background:** Klaus Havelund, John Penix and Willem Visser organized the 7th International SPIN workshop, for the first time with special focus on software model checking, by which is meant the application of model checking to code rather than to just high level designs or hardware, at Stanford University in year 2000. Model checking is a technique for testing concurrent non-deterministic systems, by efficiently exploring all possible execution paths in the system. This helps finding intermittent concurrency related errors, which conventional testing may not find. Havelund and Visser were invited as special guest editors for a special October 2002 issue of the STTT journal (International Journal on Software Tools for Technology Transfer) containing a selection of best papers from the conference. As part of this job they wrote a guest editor article, *"Program Model Checking as a New Trend"* that gives their views and experiences in the field of software model checking. Havelund and Visser are generally considered to be pioneers in this field. Two versions of the Java PathFinder tool that model check Java programs, and that have been developed in the ASE group by the POCs, have considerably influenced the international research community in this new field.
- **Accomplishment: The** research has progressed from hand translations of code to be analyzed into the language of the model checker SPIN, to the construction of JPF1, an automatic translator that handles part of Java, to the custom-made model checker JPF2, which handles the entire Java language. Model checking of large programs requires auxiliary techniques such as code abstraction and heuristics. Several such have been implemented in JPF2. For handling very large programs JPaX was developed, which analyzes events emitted by code written in any language, including C and C++. JPaX explores single execution traces and extrapolates from observations of one trace to expected properties about the program. Thus they have created a series of tools of increasing power and applicability.
- **Future Work:** JPF2 as well as JPaX are continuing research projects. The main focus of the JPF2 project is the combination of model checking and symbolic execution, where a program is executed with symbolic values. The JPaX project studies extensions of the system to detect an increasing number of errors, for example broadening the forms of deadlocks and data races that can be detected. A new effort attempts to combine the two tools, including defining a common specification logic based on temporal logic.

ASE Group Program V&V History 1997-2002



Found 5 errors in the Remote Agent using the SPIN model checker.
Program hand-translated to a model used by the SPIN model checker.

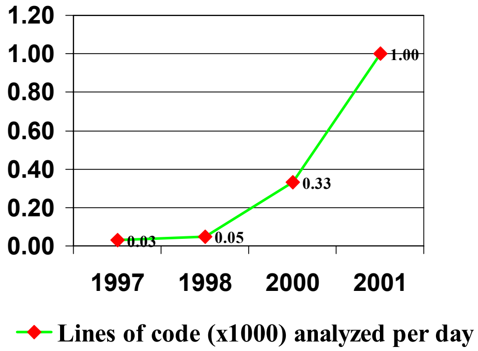
JPF 1

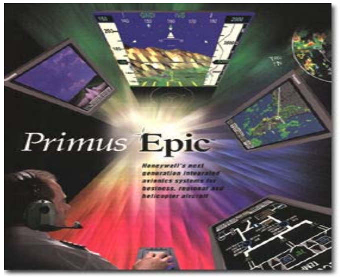
Model Checker that automatically translates Java code to a model used by the SPIN model checker.
Translates 80% of Java.

JPF 2

Custom made model checker for Java programs.
Complete coverage of Java.

1997 1998 1999 2000




Primus Epic
Honeywell's next generation integrated avionics systems for business, regional and helicopter aircraft

Discovered a critical error in the DEOS operating system, written in C, using the SPIN model checker.
The error had not been found during testing.

JPaX

Tool for monitoring temporal behavior and finding concurrency-errors (such as deadlocks and data-races) during execution of Java programs.
Applies also to C and C++ programs